Postel

NETWORK WORKING GROUP
REQUEST FOR COMMENTS #283
NIC #8165
CATEGORIES: D
OBSOLETES: NONE
UPDATES: RFC #189

R. T. BRADEN
UCLA/CCN
DECEMBER 20, 1971

NETRJT -- Remote Job Service Protocol for TIPS

A. INTRODUCTION

TIP's have very limited processing capability; their function is mainly limited to interfacing printer-keyboard devices to the Network using TELNET protocol. It will also be possible to have a tape drive on a TIP, using a subset of the count form of DTP (see RFC #264). However, TIP's cannot and will not support either DTP or FTP (see RFC #265) in general. Therefore, TIP users are exluded from using any existing remote job entry protocol (e.g. CCN's NETRJS — see RFC #189).

It appears, however, that it may be feasible in the future to use TIP's for remote job entry in one or more of the following three ways:

(a) Attach local card readers, line printers, and card punches directly to TIP ports. These devices would use a TELNET-like* format and frame their characters with Start/Stop bits. BBN can now supply a suitable 200 LPM printer, and is searching for suitable readers and punches.

(b) Connect a remote batch terminal to a full-duplex TIP port via a communication line. BBN is looking into this.

(c) Use the tape drive, and do card-to-tape and/or tape-to-print on another computer.

BBN hopes to make case (b) look exactly like (a) to the server host. That is, the remote batch terminal will send to and receive from the server in a TELNET-like format*; the printer, card reader, punch, and operator console connections will all use different sockets but one hardware port at the TIP, which will map multiple sockets into the one port.

NOTE: By "TELNET-like format", we mean: (a) CR LF used to delimit logical records (lines or cards), and (b) the ASCII or EBCDIC format effector control characters used for carriage control in the printer stream. It does *not* necessarily imply ASCII character codes.

This document describes NETRJT, a modification of CCN's NETRJS protocol specifically to provide remote job entry service to TIP's using one of the methods (a), (b), or (c). NETRJT follows the general model of NETRJS: use TELNET protocol over a primary or "operator" connection pair, and open simplex secondary connections for data transfer of job stream input and output. (We also considered the possibility of using the Divert Output mechanism of the TIP for sending remote job output over the operator connection, and an analogous mechanism for input. However, in discussion with Alex McKenzie, it was agreed that sharing the operator connections has little merit and causes lots of problems).

NETRJT differs in two principal ways from NETRJS:

1. The NETRJT server process initiates the data transfer connections, under control of commands from the remote operator console. On the other hand, under NETRJS the remote user process has responsibility for initiating the opening of secondary data transfer connections; the NETRJS server simply listens on these sockets.

2. NETRJT provides the TELNET-like format defined above for data transfer, as well as the TIP-tape DTP format. NETRJS, on the other hand, is restricted to counts to delimit logical records within DTP-like transactions, and ASA carriage control.

There are some other minor differences. For example, (1) the NETRJT server takes responsibility for folding output records when they exceed a size specified by a user command; under NETRJS, this was the user process' responsibility. (2) There are NETRJT operator commands to set the record format, record size, and code for each data transfer connection. NETRJS made the first two fixed properties of a particular terminal id, and determined the last by the choice of ICP socket. These differences imply remote operator commands in NETRJT in addition to those of NETRJS. The operator must be able to (1) cause NETRJT to open a secondary connection to a TIP socket, and (2) specify the data transfer protocol, maximum logical record length, and/or transmission code. These NETRJT commands are discussed in the following section.

CCN plans to proceed with implementation of a NETRJT server with the goal of completing an initial version by March 15, 1972. This initial version may support only DTP=BS or TT, and RECFM=TELNET or RECORDS; other options will be added as the need arises. We welcome comments and suggestions.

In the longer term, we believe that the NETRJT protocol described here should be considred as the first draft of a Network standard for remote job entry via TIP's. In its present form, NETRJT owes much to the ideas and comments of Alex McKenzie (BBN), Jon Postel (NMC), Jim White (UCSB), and Steve Wolfe (CCN).

B. NETRJT COMMANDS

NETRJT provides the following commands over the remote operator connection, in addition to the NETRJS operator commands (see Appendix D of RFC #189). The symbol "␣" denotes one or more spaces. We will use the IBM meta-language to describe the command syntax. The literal text shown here in upper case may, in fact, be entered in either upper or lower case.

1. Opening a Stream

$$\mathrm{O}\left[\mathrm{PEN}\right]\text{␣}\left\{\begin{matrix}\mathrm{PR}\ [\mathrm{INTER}]\\ \mathrm{PU}\ [\mathrm{NCH}]\\ \mathrm{R}\ [\mathrm{EADER}]\end{matrix}\right\}\left[\begin{matrix}(\text{jobname})\\ (*)\end{matrix}\right]\left[=\underset{\text{pathname}}{\text{socket-number}}\ \left[/\text{host-name}\right]\right]$$

If the specified device does not already have an open connection, the NETRJT server will request connection to the specified socket. The optional "(jobname)" parameter is used to specify a particular job by name; for more information on the semantics of this parameter, see the discussion of input and output operations below. The "/host-name" parameter, to be implemented later, is intended to allow the file to be at a host different from both user and server hosts. We include it here only to suggest a syntax.

The socket number may have a one-letter suffix D, H, or O to mean decimal, hex, or octal. Octal is the default, so the O suffix may be omitted. If BBN establishes standardized TIP sockets for specifc unit record devices, the socket number parameter could be omitted when the standard socket number is intended.

2. Closing a Stream

$$\mathrm{CL}\left[\mathrm{OSE}\right]\left[\begin{matrix}\text{␣}\ \mathrm{PR}\ [\mathrm{INTER}]\\ \text{␣}\ \mathrm{PU}\ [\mathrm{NCH}]\\ \text{␣}\ \mathrm{R}\ [\mathrm{EADER}]\end{matrix}\right]\left[,\mathrm{A}\ \left[\mathrm{CCEPT}\right]\right]$$

This command closes the specified data transfer connection. The ACCEPT option is used to signal the server that it may discard output it has transmitted, or that it has received a complete stack of job input. See discussion in next section. The device specification (PR, PU, or R) may be omitted if only one device stream is currently open.

3. Setting Format and Device Characteristics

In each of the following variants of the RJT commands, the parameter "device" is one of "PR[INTER]", "PU[NCH]", or "R[EADER]".

$$\text{RJT}\sqcup\text{D[TP] (device)} = \left\{\begin{array}{l} \text{B[S]} \\ \text{T[T]} \\ \text{D[TP]} \end{array}\right\}$$

BS: an unstructured byte stream.

TT: the TIP-tape transfer protocol (essentially the count form of Network DTP).

DTP: the Network standard DTP, complete with a modes-available handshake. This form is not useful for TIP's but is included here in acticipation of the general Network standard RJE protocol.

$$\text{RJT}\sqcup\text{R[ECFM] (device)} = \left\{\begin{array}{l} \text{T[ELNET]} \\ \text{A[SA]} \\ \text{R[ECORDS]} \\ \text{C[OMPRESSED]} \end{array}\right\}$$

The following choice of options is tentative, as it is presently unclear just what record formats will be useful for TIP tapes or remote batch terminals connected to TIP's.

TELNET:	the "TELNET-like format": CR LF used to delimit logical records in all streams, and format effector control characters (CR, LF, FF) for printer carriage control.
ASA:	CR LF used to delimit logical records, but an ASA carriage control character is sent as the first character of each printer record. (This option may be useful for remote batch terminals which expect ASA carriage control).
RECORDS:	the "truncated" format of NETRJS: an id byte, a count byte, and then the string, with ASA carriage control in each printer record.
COMPRESSED:	the "compressed" format of NETRJS (see RFC #189 for details). (Compression will be useful for batch terminals connected remotely to TIP's).

RJT␣SIZE (device) = integer

This command sets the maximum logical record length for the specified device. NETRJT will automatically fold any records exceeding this size. Default sizes are:

PR: 120

PU: 80

R : 80

RJT␣CODE (device) = $\left\{\begin{array}{l}\text{A [SCII]}\\ \text{E [BCDIC]}\end{array}\right\}$

This command sets the code to be used.

C. USING NETRJT AT CCN

1. Getting Started

a. Perform ICP to server TELNET (socket 1).

b. Execute command "RJT", yielding ready message from NETRJT.

c. Issue RJS SIGNON command.

d. These steps result in a standard full-duplex TELNET connection for an RJS remote operator console. The user can issue commands to learn the status of his jobs, send messages, reroute and abort jobs, etc.

2. Retrieving Output

a. The TIP user captures a local output device and then executes the NETRJT OPEN command for the PRINTER or PUNCH. For example, if the connection is not yet open, then either of:

```
O  PR=socket

O  PR(*)=socket
```

opens a printer connection and selects the first job in the printer queue for this terminal id.

```
O  PR(jobname)=socket
```

similarly opens a connection but selects a specified job's output. In either case, if output is not yet available the connection remains open but idle, and the output is sent when it does appear. If the socket number is omitted and the connection is not yet open, the server will prompt for a socket number.

b. If the specified output device already has an open connection, either of the open commands:

```
O   PR

O   PR(jobname)
```

may be issued to <u>accept</u> (see e. below) the last job's output and select and send another job's output. If the connection is already open, the open command may still specify "=socket", but if the specified socket does not match that currently open there will be an error message.

c. While the output stream is idle, the user can issue RJT commands with DTP, RECFM, CODE and/or SIZE parameters.

d. When the specified output is available, the server will send a stream of print line (or punched card) images. The user may issue the following RJS stream control commands (see NIC 7182 and 7183 for more information on RJS commands).

   1. BAKSPACE: repeats roughly the last page of printed output.

   2. RESTART: restarts output at the beginning of the current SYSOUT data set, or (",JOB" option) at the beginning of the job.

   3. CANCEL: deletes rest of current SYSOUT data set, or (",JOB" option) the entire job except accounting information.

   4. DEFER: stops transmission of the current job and returns it to the queue, marked "deferred". Can be restarted later, with a "backspace" ("RESET jobname" command) or from the beginning ("RESTART jobname" command).

e. The server does not discard job output until it is fully transmitted to the TIP and the user has accepted it. If the user issues a "CLOSE device" or the connection breaks accidentially (e.g. due to software or hardware failure in either host) before the output is accepted, the server saves the output with an implied BAKSPACE. When the user later reopens the connection and again selects this job (either explicitly by name or by calling for the next job), it will be retransmitted, repeating the last page. The user can also defer or restart the job output before reopening the connection. Note that CLOSE without the ACCEPT option is generally a "panic" control to stop the output stream if the printer paper jams, etc.

f. Transmitted output will be considered accepted by the user if:

   1. The user issues a new OPEN command for that device.

   2. The user issues a "CLOSE device, ACCEPT" (e.g. "CL␣PR,A") command for that device. This command will be held pending until job output in progress has completed. After the last RFNM is received, the connection will be closed and the job output discarded at the server end.

3. The original OPEN command specified "(*)", i.e. an asterisk for jobname. This implies that the device stream is going to be running continuously and that the user does not want to explicitly request each output job or to accept each one. Thus, if the stream is opened "(*)", then the server assumes each job is accepted when the RFNM returns from the last block.

3. Sending Input

(a) The user sends the following remote operator command to the server:

OPEN READER=socket

(This may be shortened to "O R=socket"). The server will send an RFC to the user's card reader socket on which his TIP should be listening. The server will issue an operator message when the connection is open. The connection will be considered idle until the first card image is received by the server. The OPEN command will be ignored if the connection is already open, or if an earlier open request is pending.

(b) Before or after the open, but while the connection is idle, the user may issue RJT commands to set the record format, data transfer protocol, code, and/or maximum record size to different values.

(c) The user sends in a stream of card images which constitute one or more jobs. The server will discard the spooled images for a job without processing them if the user issues a CANCEL READER command or if the connection breaks (e.g. due to software or hardware failures in either host) before the job is accepted for processing. The stream then becomes idle again.

(d) A spooled job will be accepted by the server only when one of the following occurs:

1. A server-dependent end-of-job card (e.g. "//null" at CCN) is received by the server. The last job is accepted, and the stream becomes idle until another card is received.
2. A server-dependent beginning-of-job card (e.g. a "JOB" card at CCN) is received by the server. The previous job is accepted but the stream does not become idle at this time.
3. The user issues a CLOSE READER,ACCEPT (or "CL R,A") command to the server. The stream is closed.

(e) The user can issue a CLOSE READER ("CL R") command to close the stream. However, this command will be held pending by the server until the stream is idle, unless the form "CLOSE READER,ACCEPT" is issued. A CLOSE will cancel a pending OPEN command, and vice versa. The server will send the remote operator a message when a connection opens or closes.

(f) Some servers (e.g. CCN) will extract the jobname for each input job from the reader stream. However, the OPEN command may specify a particular jobname, overriding that in the reader stream. That is, the jobname from the OPEN command will replace that appearing in the first job of the newly-opened stream. This feature is merely a convenience, and was included mainly for syntactic consistency between input and output. However, the use of asterisk as a jobname has no meaning for the reader stream, and will be ignored.

ARPA NETWORK
NIC 81 65
INFORMATION CENTER

NETWORK WORKING GROUP
REQUEST FOR COMMENTS #283
NIC #8165
CATEGORIES: D
OBSOLETES: NONE
UPDATES: RFC #189

R. T. BRADEN
UCLA/CCN

DECEMBER 9, 1971

NETRJT -- Remote Job Service Protocol for TIPs

A. INTRODUCTION

TIP's have very limited processing capability; their function is mainly limited to interfacing printer-keyboard devices to the Network using TELNET protocol. It will also be possible to have a tape drive on a TIP, using something like the count form of DTP (see RFC #264); however, the details of this TIP-tape protocol are not yet available from BBN. In any case, TIP's cannot and will not support either DTP or FTP (see RFC #265) in general. Therefore, TIP users are excluded from using the NETRJS remote job entry protocol (see RFC #189).

It appears, however, that it may be feasible in the future to use TIP's for remote job entry in one or more of the following three ways:

(a) Attach local card readers and line printers directly to TIP ports. These devices would use a TELNET-like* format and frame their characters with Start/Stop bits.

(b) Connect a remote batch terminal to a full-duplex TIP port via a communication line.

(c) Use the tape drive, and do card-to-tape and/or tape-to-print on another computer.

It is hoped that case (b) can be made to look exactly like (a) to the server host. That is, the remote batch terminal will send to and receive from the server in a TELNET-like format*; the printer, card reader, and two operator console connections will all use different sockets but one hardware port at the TIP.

NOTE: By "TELNET-like format", we mean: (a) logical records (lines or cards) would be delimited by CR LF, and (b) the ASCII (or EBCDIC) format effector control characters would be used for carriage control in the printer stream. It does *not* necessarily imply ASCII character codes.

This document describes NETRJT, a modification of CCN's NETRJS protocol specifically to provide remote job entry service to TIP's using one of the methods (a), (b), or (c). NETRJT follows the general model of NETRJS: use TELNET protocol over a primary or "operator" connection pair, and open simplex secondary connections for data transfer of job stream input and output. (We also considered the possibility of using the Divert Output mechanism of the TIP for sending remote job output over the operator connection, and an analogous mechanism for input. However, in discussion with Alex McKenzie, it was agreed that sharing the operator connections has little merit and causes lots of problems).

NETRJT differs in two principal ways from NETRJS:

1. The NETRJT server process initiates the data transfer connections, under control of commands from the remote operator console. On the other hand, under NETRJS the remote user process has responsibility for initiating the opening of secondary data transfer connections; the NETRJS server simply listens on these sockets.
2. NETRJT uses the TELNET-like format defined above for data transfer (or else the still-unspecified TIP-tape format). NETRJS, on the other hand, uses counts to delimit logical records within DTP-like transactions, and also uses ASA carriage control.

There are some other more minor differences. For example, the NETRJT server takes responsibility for folding output records when they exceed a size specified by a user command; under NETRJS, this was the user process' responsibility. There are NETRJT operator commands to set the record format, record size, and code for each data transfer connection. NETRJS made the first two fixed properties of a particular terminal id, and determined the last by the choice of ICP socket. These differences imply remote operator commands in NETRJT in addition to those of NETRJS. The operator must be able to (1) cause NETRJT to open a secondary connection to a TIP socket, (2) specify either TELNET-like format or TIP-tape protocol, (3) specify maximum logical record length, (4) specify transmission code. These NETRJT commands are discussed in the following section.

CCN plans to proceed with implementation of a NETRJT server with the goal of completing an initial version by March 1, 1972. This initial version may support only DTP=TELNET and RECFM=TELNET; other options will be added as the need arises. We welcome comments and suggestions.

In the longer term, we believe that the NETRJT protocol described here should be considered as the first draft of a Network standard for remote job entry via TIP's. In its present form, NETRJT owes much to the ideas and comments of Alex McKenzie (BBN), Jon Postel (NMC), Jim White (UCSB), and Steve Wolfe (CCN).

B. NETRJT COMMANDS

NETRJT provides the following commands over the remote operator connection, in addition to the NETRJS operator commands (see Appendix D of RFC #189). The symbol "␣" denotes one or more spaces. We will use the IBM meta-language to describe the command syntax. The literal text shown here in upper case may, in fact, be entered in either upper or lower case.

1. Opening a Stream

$$\text{O}\left[\text{PEN}\right]\sqcup\left\{\begin{array}{l}\text{PR}\left[\text{INTER}\right]\\ \text{PU}\left[\text{NCH}\right]\\ \text{R}\left[\text{EADER}\right]\end{array}\right\}\left[\begin{array}{c}\text{(jobname)}\\ \text{(*)}\end{array}\right]\left[\text{=socket-number}\left[\text{@host-name}\right]\right]$$

If the specified device does not already have an open connection, the NETRJT server will request connection to the specified socket. The optional "(jobname)" parameter is used to specify a particular job by name; for more information on the semantics of this parameter, see the discussion of input and output operations below. The "@host-name" parameter, to be implemented later, is intended to allow the file to be at a host different from both user and server hosts. We include it here only to suggest a syntax.

The socket number may have a one-letter suffix D, H, or O to mean decimal, hex, or octal. Octal is the default, so the O suffix may be omitted. If BBN establishes standardized TIP sockets for specifc unit record devices, the socket number parameter could be omitted when the standard socket number is intended.

2. Closing a Stream

$$\text{CL}\left[\text{OSE}\right]\left[\begin{array}{l}\sqcup\,\text{PR}\left[\text{INTER}\right]\\ \sqcup\,\text{PU}\left[\text{NCH}\right]\\ \sqcup\,\text{R}\left[\text{EADER}\right]\end{array}\right]\left[\text{,A}\left[\text{CCEPT}\right]\right]$$

This command closes the specified data transfer connection. The ACCEPT option is used to signal the server that it may discard output it has transmitted, or that it has received a complete stack of job input. See discussion in next section. The device specification (PR, PU, or R) may be omitted if only one device stream is currently open.

3. Setting Format and Device Characteristics

In each of the folloing variants of the RJT commands, the parameter "device" one of "PR [INTER]", "PU [NCH]", or "R[EADER]".

$$\text{RJT⌴DTP} \quad \text{(device)} = \left\{ \begin{array}{l} \text{T [ELNET]} \\ \text{C [OUNT]} \end{array} \right\}$$

This command determines the data transfer protocol to be used on specified device. The options are:

TELNET:	means an ASCII byte stream is sent.
COUNT:	means the (unspecified) data transfer protocol for TIP tapes is to be used.

$$\text{RJT⌴RECFM} \quad \text{(device)} = \left\{ \begin{array}{l} \text{T [ELNET]} \\ \text{A [SA]} \\ \text{C [OUNT]} \end{array} \right\}$$

This command determines the record format to be used. The options are:

TELENT:	means ASCII format effectors in PRINTER streams, and CR LF to delimit logical records in all streams; this is the default.
ASA:	means CR LF to delimit logical records, but ASA carriage control characters are sent in PRINTER stream. This option may be useful for remote batch terminals which use ASA carriage control, or TIP-tapes to be printed off-line.
COUNT:	is the "truncated" format of NETRJS, using counts and strings.

RJT⌴SIZE (device) = integer

This command sets the maximum logical record length for the specified device. NETRJT will automatically fold any records exceeding this size. Default sizes are:

PR	120
PU	80
R	80

$$\text{RJT}\sqcup\text{CODE (device)} = \left\{ \begin{array}{l} \text{A} \left[\text{SCII}\right] \\ \text{E} \left[\text{BCDIC}\right] \end{array} \right\}$$

This command sets the code to be used.

C. USING NETRJT AT CCN

1. Getting Started

a. Perform ICP to server TELNET (socket 1).

b. Execute command "RJT", yielding ready message from NETRJT.

c. Issue RJS SIGNON command.

d. These steps result in a standard full-duplex TELNET connection for an RJS remote operator console. The user can issue commands to learn the status of his jobs, send messages, reroute and abort jobs, etc.

2. Retrieving Output

a. The TIP user captures a local output device and then executes the NETRJT OPEN command for the PRINTER or PUNCH. For example, if the connection is not yet open, then either of:

```
O  PR=socket

O  PR(*)=socket
```

opens a printer connection and selects the first job in the printer queue for this terminal id.

```
O  PR(jobname)=socket
```

similarly opens a connection but selects a specified job's output. In either case, if output is not yet available the connection remains open but idle, and the output is sent when it does appear. If the socket number is omitted and the connection is not yet open, the server will prompt for a socket number.

b. If the specified output device already has an open connection, either of the open commands:

```
O   PR

O   PR(jobname)
```

may be issued to accept (see e. below) the last job's output and select and send another job's output. If the connection is already open, the open command may still specify "=socket", but if the specified socket does not match that currently open there will be an error message.

c. While the output stream is idle, the user can issue RJT commands with DTP, RECFM, CODE and/or SIZE parameters.

d. When the specified output is available, the server will send a stream of print line (or punched card) images. The user may issue the following RJS stream control commands (see NIC 7182 and 7183 for more information on RJS commands).

   1. BAKSPACE: repeats roughly the last page of printed output.

   2. RESTART: restarts output at the beginning of the current SYSOUT data set, or (",JOB" option) at the beginning of the job.

   3. CANCEL: deletes rest of current SYSOUT data set, or (",JOB" option) the entire job except accounting information.

   4. DEFER: stops transmission of the current job and returns it to the queue, marked "deferred". Can be restarted later, with a "backspace" ("RESET jobname" command) or from the beginning ("RESTART jobname" command).

e. The server does not discard job output until it is fully transmitted to the TIP and the user has accepted it. If the user issues a "CLOSE device" or the connection breaks accidentially (e.g. due to software or hardware failure in either host) before the output is accepted, the server saves the output with an implied BAKSPACE. When the user later reopens the connection and again selects this job (either explicitly by name or by calling for the next job), it will be retransmitted, repeating the last page. The user can also defer or restart the job output before reopening the connection. Note that CLOSE without the ACCEPT option is generally a "panic" control to stop the output stream if the printer paper jams, etc.

f. Transmitted output will be considered accepted by the user if:

   1. The user issues a new OPEN command for that device.

   2. The user issues a "CLOSE device, ACCEPT" (e.g. "CL␣PR,A") command for that device. This command will be held pending until job output in progress has completed. After the last RFNM is received, the connection will be closed and the job output discarded at the server end.

   4.

3. The original OPEN command specified "(*)", i.e. an asterisk for jobname. This implies that the device stream is going to be running continuously and that the user does not want to explicitly request each output job or to accept each one. Thus, if the stream is opened "(*)", then the server assumes each job is accepted when the RFNM returns from the last block.

g. The user can issue a CLOSE command to the printer (CL PR") or punch ("CL PU"). This command will be held pending by the server until the stream is idle, i.e. no job is in the process of being transmitted. Closing an output stream always implies "ACCEPT", so the option need not be specified.

3. Sending Input

(a) The user sends the following remote operator command to the server:

```
OPEN    READER=socket
```

(This may be shortened to "O R=socket"). The server will send an RFC to the user's card reader socket on which his TIP should be listening. The server will issue an operator message when the connection is open. The connection will be considered idle until the first card image is received by the server. The OPEN command will be ignored if the connection is already open, or if an earlier open request is pending.

(b) Before or after the open, but while the connection is idle, the user may issue RJT commands to set the record format, data transfer protocol, code, and/or maximum record size to different values.

(c) The user sends in a stream of card images which constitute one or more jobs. The server will discard the spooled images for a job without processing them if the user issues a CANCEL READER command or if the connection breaks (e.g. due to software or hardware failures in either host) before the job is accepted for processing. The stream then becomes idle again.

(d) A spooled job will be accepted by the server only when one of the following occurs:

1. A server-dependent end-of-job card (e.g. "//null" at CCN) is received by the server. The last job is accepted, and the stream becomes idle until another card is received.
2. A server-dependent beginning-of-job card (e.g. a "JOB" card at CCN) is received by the server. The previous job is accepted but the stream does not become idle at this time.
3. The user issues a CLOSE READER,ACCEPT (or "CL␣R,A") command to the server. The stream is closed.

(e) The user can issue a CLOSE READER ("CL␣R") command to close the stream. However, this command will be held pending by the server until the stream is idle, unless the form "CLOSE READER,ACCEPT" is issued. A CLOSE will cancel a pending OPEN command, and vice versa. The server will send the remote operator a message when a connection opens or closes.

(f) Some servers (e.g. CCN) will extract the jobname for each input job from the reader stream. However, the OPEN command may specify a particular jobname, overriding that in the reader stream. That is, the jobname from the OPEN command will replace that appearing in the first job of the newly-opened stream. This feature is merely a convenience, and was included mainly for syntactic consistency between input and output. However, the use of asterisk as a jobname has no meaning for the reader stream, and will be ignored.